



گزارش پروژه تشخیص چهره انسان

استاد: مهندس علی تورانی

اعضای گروه: مانا ناظمی ۹۶۰۱۲۲۶۸۰۰۷۸

میلاد گلیج ۹۶۰۱۲۲۶۸۰۰۰۲

مقدمه:

همانطور که همه ی ما می دانیم تشخیص چهره موجودات برای ما کاملاً قابل فهم است، ما می توانیم گربه، سگ، و هر موجود دیگری را از انسان تشخیص دهیم. اما هوش مصنوعی توانایی درک این را ندارد که خودش بتواند تفاوت هارا متوجه شود و تشخیص دهد که چهره متعلق به انسان است یا خیر.

در این پروژه ما بررسی می کنیم که چگونه میشود به هوش مصنوعی این را بگوییم که چهره انسان را از باقی موجودات تشخیص دهد. با کمک زبان پایتون این موضوع را بررسی کردیم و به نتایجی رسیدیم که در ادامه مراحل را بررسی میکنیم.

در ابتدای کد دیتاست های مورد نیاز پروژه را در گوگل درایو خود قرار داده و سپس در کولب ایمپورت میکنیم. در ابتدا مسیر پوشه **person** که همان دیتاست انسان می باشد را به کد می شناسونیم، که قرار است ۹ تصویر نشان دهیم پس یک شبکه 3×3 از تصاویر انسان درست میکنیم. (مسیر فایل عکس های انسان را باید قرار دهیم (**person**)) در تصویر زیر مشاهده میکنیم:

```
for i in range(9):
    # define subplot
    pyplot.subplot(330 + 1 + i)
    #we will give it the file name to read
    filename = folder + 'person_000' + str(i) + '.jpg'
    #load the pixels into the image
    image = imread(filename)
    #make the pixel datas
    pyplot.imshow(image)
#show the image
pyplot.show()
```

در بخش بعد همان کاری که برای نمایش ۹ تایی تصاویر انسان انجام دادیم را برای سگ و گربه انجام می‌دهیم و به جای آدرس پوشه انسان، سگ و گربه را قرار می‌دهیم:

برای تصاویر سگ:

```
for i in range(9):
    # define subplot
    pyplot.subplot(330 + 1 + i)
    #we will give it the file name to read
    filename = folder + 'dog_000' + str(i) + '.jpg'
    #load the pixels into the image
    image = imread(filename)
    #make the pixel datas
    pyplot.imshow(image)
#show the image
pyplot.show()
```

برای تصاویر گربه:

```
for i in range(9):
    # define subplot
    pyplot.subplot(330 + 1 + i)
    #we will give it the file name to read
    filename = folder + 'cat_000' + str(i) + '.jpg'
    #load the pixels into the image
    image = imread(filename)
    #make the pixel datas
    pyplot.imshow(image)
#show the image
pyplot.show()
```

بعد از نمایش و معرفی مسیر تصاویر و دیتاست ها حال باید کتابخانه های مورد نیاز را import کنیم:

Os → Directory ساخت برای

Listdir → Directory برای خواندن

Asarray → برای تبدیل به آرایه

Save → برای سیو تصویر

```
[ ] import os
    from os import listdir
    from numpy import asarray
    from numpy import save
```

سپس کتابخانه های کراس را import میکنیم:

Load img → برای خواندن تصاویر

Img to array → برای تبدیل تصویر به آرایه

Save img → برای سیو تصویر

```
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import save_img
```

حال باید اندازه تصاویر را یک اندازه کنیم (تمامی تصاویر به ۲۵۶)

اینکار را برای تصاویر سگ و گربه هم انجام میدهیم:

برای انسان:

```
[ ] folder = '/content/drive/My Drive/Deep/person/'
    reshape_dir = "/content/drive/My Drive/Deep/reshape_human/"
    os.mkdir(reshape_dir)
    #define two lists for holdin gthe photos array and labels array
    #images, labels = list(), list()
    i = 0;
    #this for will look up the file in the directory above
    for file in listdir(folder):
        i = i+1
        print('%0.4f percent**** \n' % ((i+1)/985*100))
        #label the classes, 1 for human and 0 for non-human(cats and dogs)
        out_label = 1.0
        #load the images
        image = load_img(folder + file, target_size=(256,256))
        #convert the image to a numpy array
        image = img_to_array(image)
        save_img(reshape_dir + file, image)
        #add the image to images list and append it's label to labels list
        #images.append(image)
        #labels.append(out_label)
```

برای سگ:

```
[ ] #Now we load the non-human(dogs and cats) images
    folder = '/content/drive/My Drive/Deep/dog/'
    reshape_dir = "/content/drive/My Drive/Deep/reshape_dog/"
    os.mkdir(reshape_dir)
    #this for will look up the file in the directory above
    for file in listdir(folder):
        print(file + ' has been reshaped\n')
        #label the classes, 1 for human and 0 for non-human(cats and dogs)
        out_label = 0.0
        #load the images
        image = load_img(folder + file, target_size=(256, 256))
        #convert the image to a numpy array
        image = img_to_array(image)
        save_img(reshape_dir + file, image)
        #add the image to images list and append it's label to labels list
        #images.append(image)
        #labels.append(out_label)
```

برای گریه:

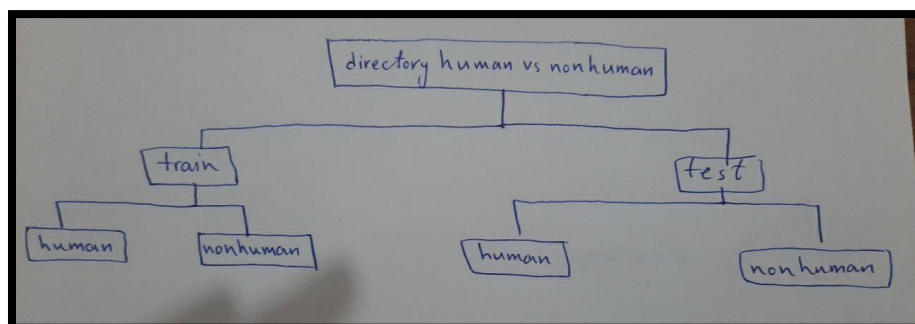
```
folder = '/content/drive/My Drive/Deep/cat/'
reshape_dir = "/content/drive/My Drive/Deep/reshape_cat/"
os.mkdir(reshape_dir)
#this for will look up the file in the directory above
for file in.listdir(folder):
    print(file + ' has been reshaped\n')
    #label the classes, 1 for human and 0 for non-human(cats and dogs)
    out_label = 0.0
    #load the images
    image = load_img(folder + file, target_size=(256, 256))
    #convert the image to a numpy array
    image = img_to_array(image)
    save_img(reshape_dir + file, image)
    #add the image to images list and append it's label to labels list
    #images.append(image)
    #labels.append(out_label)
```

کراس ویژگی که داره میتونه اسم پوشه هارو به عنوان دسته بندی train and test در

نظر بگیره و ما هم میخواییم یک دایرکتوری درست کنیم به اسم human vs

unhuman و میخواییم اینو به دو دسته train و test تقسیم کنیم، که train و test

هرکدام به دو دسته تقسیم میشوند:



Test و Train دایرکتوری ها

```
[ ] #now we train and test the model
def run_model():
    # define model
    model = primary_model()
    # create data generator
    imageGEN = ImageDataGenerator(rescale=1.0/255.0)
    # prepare iterators
    train_it = imageGEN.flow_from_directory('/content/drive/My Drive/Deep/human_vs_nonhuman/train',
        class_mode='binary', batch_size=16, target_size=(256, 256))
    test_it = imageGEN.flow_from_directory('/content/drive/My Drive/Deep/human_vs_nonhuman/test',
        class_mode='binary', batch_size=4, target_size=(256, 256))
    # fit the model
    history = model.fit_generator(train_it, steps_per_epoch=len(train_it),
        validation_data=test_it, validation_steps=len(test_it), epochs=8, verbose=0)
    #see the accuracy
    _, acc = model.evaluate_generator(test_it, steps=len(test_it), verbose=0)
    print('Accuracy : %.4f' % (acc * 100.0))
    model.save('/content/drive/My Drive/Deep/final_model.h5')
    return history
```

```
[ ] dataset_dir = "/content/drive/My Drive/Deep/human_vs_nonhuman/"
os.mkdir(dataset_dir)
subdirs = ["train/", "test/"]
for subdir in subdirs:
    # create label subdirectories
    path1 = os.path.join(dataset_dir, subdir)
    os.mkdir(path1)
    subdirsname = ["human", "nonhuman"]
    for dirname in subdirsname:
        path = os.path.join(path1, dirname)
        os.mkdir(path)
```

حال پوشه هامون (path) رو که ساختیم ، باید reshape هامون رو ببریم داخل اون پوشه ها (path) بریزیم و برای اینکار از کتابخونه های مقابل استفاده میکنیم: shutil و

random

اینکار را هم برای فایل reshape human ، reshape dog و reshape cat

انجام میدهیم:

```
[ ] from shutil import copyfile
    from random import seed
    from random import random
    #make a random number
    seed(1)
    #15 % of images are for test
    val_ratio = 0.15
    # copy training dataset images into subdirectories
    src_directory = '/content/drive/My Drive/Deep/reshape_human'
    for file in listdir(src_directory):
        src = src_directory + '/' + file
        dst_dir = 'train/'
        if random() < val_ratio:
            dst_dir = 'test/'
        dst = dataset_dir + dst_dir + 'human/' + file
        copyfile(src, dst)
```

Cat & Dog

```
[ ] #make a random number
    seed(1)
    #15 % of images are for test
    val_ratio = 0.15
    # copy training dataset images into subdirectories
    src_directory = '/content/drive/My Drive/Deep/reshape_dog'
    for file in listdir(src_directory):
        src = src_directory + '/' + file
        dst_dir = 'train/'
        if random() < val_ratio:
            dst_dir = 'test/'
        dst = dataset_dir + dst_dir + 'nonhuman/' + file
        copyfile(src, dst)
    seed(1)
    src_directory = '/content/drive/My Drive/Deep/reshape_cat'
    for file in listdir(src_directory):
        src = src_directory + '/' + file
        dst_dir = 'train/'
        if random() < val_ratio:
            dst_dir = 'test/'
        dst = dataset_dir + dst_dir + 'nonhuman/' + file
        copyfile(src, dst)
```

حال باید شبکه deep خودمون را بسازیم، ابتدا کتابخانه های مورد نیاز را import میکنیم:


```
[ ] #the librarie sthat we need to build a primary model as reference of future comparisons
import sys
from matplotlib import pyplot
from keras.utils import to_categorical
from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Dense
from keras.layers import Flatten
from keras.optimizers import SGD
from keras.preprocessing.image import ImageDataGenerator
```

حال یک تابع تعریف میکنیم و میگیریم که مدلون sequential (لایه لایه) هستش و همینطور بصورت لایه لایه deep رو میسازیم: (شبکه عصبیون رو به صورت لایه لایه میسازیم)

```
def primary_model():
    model = Sequential()
    model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(256, 256, 3)))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(256, 256, 3)))
    model.add(MaxPooling2D((2, 2)))
    model.add(Flatten())
    model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
    model.add(Dense(1, activation='sigmoid'))
```

حال باید compile شود:

```
opt = SGD(lr=0.001, momentum=0.9)
model.compile(optimizer=opt, loss='binary_crossentropy', metrics=['accuracy'])
return model
```

بعد از compile از آن استفاده میکنیم و save میکنیم تا برای استفاده مجدد زمان رسیدن به جواب سریع تر باشد. با استفاده از کتابخانه keras.model مدل را load کرده و یک تصویر برای test میخوانیم و به آرایه npy تبدیل میکنیم و پس از آن reshape میکنیم.

حال تبدیل به float32 میکنیم و بعد مدلمون رو که قبله ساخته ایم را load میکنیم و بعد از تابع predict برای کتابخانه کراس استفاده میکنیم که اگر انسان بود =۱ میشود و در غیر اینصورت =۰

این مراحل برای گربه و سگ هم به همانصورت است.

```
[ ] from keras.models import load_model

image = imread('/content/drive/My Drive/Deep/human_vs_nonhuman/test/human/person_0006.jpg')
#make the pixel datas
pyplot.imshow(image)
#show the image
pyplot.show()

# load the image
img = load_img('/content/drive/My Drive/Deep/human_vs_nonhuman/test/human/person_0006.jpg', target_size=(256, 256))

# convert to array
img = img_to_array(img)
# reshape into a single sample with 3 channels
img = img.reshape(1, 256, 256, 3)
# center pixel data
img = img.astype('float32')

# load model
model = load_model('/content/drive/My Drive/Deep/final_model.h5')
output = model.predict(img)
if(output == 0):
    print('I\'m 100% sure, He/She is a human')
else:
    print('I\'m 100% sure,it is a non-human creature')
```

خروجی:

